

Systemy operacyjne 1

Systemy plików

Arkadiusz Chrobot

Katedra Systemów Informatycznych, Politechnika Świętokrzyska w Kielcach

Kielce, 13 stycznia 2026

Plan wykładu

1 Wstęp

2 Pliki

- 1 Definicja pliku
- 1 Typ pliku
- 1 Operacje na plikach
- 1 Metody dostępu
- 1 Semantyka integralności

3 Katalogi

- 1 Katalogi fizyczne i logiczne
- 1 Wykaz pozycji
- 1 Struktury katalogów

4 Ochrona plików

Plan wykładu

❶ Wstęp

❷ Pliki

- ❶ Definicja pliku
- ❷ Typ pliku
- ❸ Operacje na plikach
- ❹ Metody dostępu
- ❺ Semantyka integralności

❸ Katalogi

- ❶ Katalogi fizyczne i logiczne
- ❷ Wykaz pozycji
- ❸ Struktury katalogów

❹ Ochrona plików

Plan wykładu

❶ Wstęp

❷ Pliki

- ❶ Definicja pliku
- ❷ Typ pliku
- ❸ Operacje na plikach
- ❹ Metody dostępu
- ❺ Semantyka integralności

❸ Katalogi

- ❶ Katalogi fizyczne i logiczne
- ❷ Wykaz pozycji
- ❸ Struktury katalogów

❹ Ochrona plików

Plan wykładu

❶ Wstęp

❷ Pliki

❶ Definicja pliku

❷ Typ pliku

❸ Operacje na plikach

❹ Metody dostępu

❺ Semantyka integralności

❸ Katalogi

❶ Katalogi fizyczne i logiczne

❷ Wykaz pozycji

❸ Struktury katalogów

❹ Ochrona plików

Plan wykładu

❶ Wstęp

❷ Pliki

❶ Definicja pliku

❷ Typ pliku

❸ Operacje na plikach

❹ Metody dostępu

❺ Semantyka integralności

❸ Katalogi

❶ Katalogi fizyczne i logiczne

❷ Wykaz pozycji

❸ Struktury katalogów

❹ Ochrona plików

Plan wykładu

❶ Wstęp

❷ Pliki

❶ Definicja pliku

❷ Typ pliku

❸ Operacje na plikach

❹ Metody dostępu

❺ Semantyka integralności

❸ Katalogi

❶ Katalogi fizyczne i logiczne

❷ Wykaz pozycji

❸ Struktury katalogów

❹ Ochrona plików

Plan wykładu

❶ Wstęp

❷ Pliki

❶ Definicja pliku

❷ Typ pliku

❸ Operacje na plikach

❹ Metody dostępu

❺ Semantyka integralności

❸ Katalogi

❶ Katalogi fizyczne i logiczne

❷ Wykaz pozycji

❸ Struktury katalogów

❹ Ochrona plików

Plan wykładu

❶ Wstęp

❷ Pliki

- ❶ Definicja pliku
- ❷ Typ pliku
- ❸ Operacje na plikach
- ❹ Metody dostępu
- ❺ Semantyka integralności

❸ Katalogi

- ❶ Katalogi fizyczne i logiczne
- ❷ Wykaz pozycji
- ❸ Struktury katalogów

❹ Ochrona plików

Plan wykładu

❶ Wstęp

❷ Pliki

- ❶ Definicja pliku
- ❷ Typ pliku
- ❸ Operacje na plikach
- ❹ Metody dostępu
- ❺ Semantyka integralności

❸ Katalogi

- ❶ Katalogi fizyczne i logiczne
- ❷ Wykaz pozycji
- ❸ Struktury katalogów

❹ Ochrona plików

Plan wykładu

- ❶ Wstęp
- ❷ Pliki
 - ❶ Definicja pliku
 - ❷ Typ pliku
 - ❸ Operacje na plikach
 - ❹ Metody dostępu
 - ❺ Semantyka integralności
- ❸ Katalogi
 - ❶ Katalogi fizyczne i logiczne
 - ❷ Wykaz pozycji
 - ❸ Struktury katalogów
- ❹ Ochrona plików

Plan wykładu

- ❶ Wstęp
- ❷ Pliki
 - ❶ Definicja pliku
 - ❷ Typ pliku
 - ❸ Operacje na plikach
 - ❹ Metody dostępu
 - ❺ Semantyka integralności
- ❸ Katalogi
 - ❶ Katalogi fizyczne i logiczne
 - ❷ Wykaz pozycji
 - ❸ Struktury katalogów
- ❹ Ochrona plików

Plan wykładu

- ❶ Wstęp
- ❷ Pliki
 - ❶ Definicja pliku
 - ❷ Typ pliku
 - ❸ Operacje na plikach
 - ❹ Metody dostępu
 - ❺ Semantyka integralności
- ❸ Katalogi
 - ❶ Katalogi fizyczne i logiczne
 - ❷ Wykaz pozycji
 - ❸ Struktury katalogów
- ❹ Ochrona plików

Plan wykładu

- ❶ Wstęp
- ❷ Pliki
 - ❶ Definicja pliku
 - ❷ Typ pliku
 - ❸ Operacje na plikach
 - ❹ Metody dostępu
 - ❺ Semantyka integralności
- ❸ Katalogi
 - ❶ Katalogi fizyczne i logiczne
 - ❷ Wykaz pozycji
 - ❸ Struktury katalogów
- ❹ Ochrona plików

Wstęp

Sposób przechowywania informacji jest specyficzny dla każdego rodzaju pamięci¹, dlatego też systemy operacyjne oferują specjalne struktury danych nazywane plikami. Pliki pozwalają uzyskać dostęp do informacji w ten sam sposób, niezależnie od tego gdzie ta informacja jest zapisana. Aby ułatwić zarządzanie plikami systemy operacyjne utrzymują informację o nich w katalogach, które z kolei są organizowane w struktury katalogów. Struktury katalogów wraz z plikami tworzą *system plików*. System plików jest najbardziej dostrzegalną dla użytkownika częścią systemu operacyjnego. Często systemy operacyjne obsługują więcej niż jeden system plików.

¹Proszę porównać np. dysk CD i dysk twardy.

Definicja pliku

Plik jest abstrakcyjnym typem danych dostarczonym przez system operacyjny w celu realizacji spójnej metody dostępu do informacji umiejscowionych w pamięciach różnego typu. Pliki posiadają szereg atrybutów, z których najważniejszym, dla użytkownika jest jednoznacznie je identyfikująca nazwa. Format pliku może być swobodny lub ściśle określony. Do plików o swobodnym formacie należą pliki tekstowe, w których dane są przechowywane w postaci ciągów znaków zakończonych znakiem(-kami) końca wiersza. Pliki o określonym formacie zawierają najczęściej numeryczne dane binarne lub rekordy danych. Ogólnie, pliki mogą przechowywać programy, zarówno w postaci źródłowej, jak i wynikowej oraz dane. Informacje w plikach umieszczane są porcjami, które nazywamy *rekordami logicznymi*. Rozmiar takiego rekordu jest zmienny, w szczególności może on wynosić jeden bajt. Ponieważ rekordy logiczne są rozmieszczane w blokach alokacji, a rzadko się zdarza, aby rozmiar rekordu logicznego był wielokrotnością rozmiaru bloku alokacji, to jest to przyczyną powstawania fragmentacji.

Typy plików

Typ pliku jest atrybutem pliku, który określa jego strukturę. Podczas projektowania systemu operacyjnego należy określić w jaki sposób i czy w ogóle będzie on interpretował typy plików. Jednym z najciekawszych przykładów systemów, które rozpoznają typ pliku jest TOPS-20. W tym systemie wraz z plikiem zawierającym kod wynikowy programu znajdowały się również pliki z kodem źródłowym. Jeśli data modyfikacji pliku wynikowego lub jego wersja była późniejsza niż plików z kodem źródłowym, to system operacyjny przed uruchomieniem programu wywoływał kompilator, który generował nowszą wersję wynikową programu. Po aktualizacji program był ostatecznie wykonywany. Pozwolenie systemowi operacyjnemu na rozpoznawanie typów plików wiąże się z koniecznością umożliwienia mu definiowania nowych typów plików. Przykładem systemów dysponujących taką własnością są systemy rodziny Windows, w których użytkownik może skojarzyć określoną akcję z konkretnym typem pliku. Inne podejście może zakładać, że system operacyjny w ogóle nie będzie interpretował typów plików. Takie podejście zastosowano w systemie Unix. Zawartość plików w tym systemie jest traktowana jako ciąg bajtów, a ich interpretacja jest pozostawiana aplikacjom użytkownika.

Operacje na plikach

Ponieważ plik jest abstrakcyjną strukturą danych, to system operacyjny musi dostarczyć definicji operacji, które mogą być na niej wykonywane. Istnieje pięć takich operacji, które są uznawane za podstawowe: *tworzenie pliku*, *zapis pliku*, *odczyt pliku*, *ustawienie wskaźnika pliku*, *usunięcie pliku*. Wszystkie te operacje wiążą się z przeszukaniem katalogu i znalezieniem pozycji odpowiadającej plikowi, na którym operacja ma zostać przeprowadzona (w przypadku tworzenia pliku musi to być pozycja pusta). Aby wyeliminować przeszukiwanie katalogu z tych operacji wprowadza się dodatkową operację nazywaną *otwieraniem pliku*. Jeśli plik jest otwierany, to system operacyjny tworzy w pamięci operacyjnej kopię jego wpisu w katalogu. Operacją uzupełniającą do otwierania jest *zamykanie pliku*, które powoduje zapisanie wpisu z pamięci operacyjnej na dysk, jeśli te wersje się różnią i usunięcie wpisu z pamięci. Niektóre systemy dokonują niejawnego otwierania i zamykania plików, ale większość wymaga jawnego użycia przez aplikację użytkownika odpowiednich wywołań systemowych. Ze wskaźnika pliku, oprócz operacji jego ustawiania korzystają również operacje odczytu i zapisu. Wartość początkowa wskaźnika danego pliku może być zapisywana w jego pozycji katalogu, ale najczęściej jest przy otwieraniu ustawiana na zero (początek pliku). Wymienione na początku operacje są operacjami podstawowymi. Część innych operacji możemy stworzyć używając tych podstawowych, jak np. kopiowanie plików. Inne jak np.: zmiana nazwy pliku muszą być wykonywane przez system operacyjny. Systemy wielozadaniowe dostarczają także operacji blokowania (ang. *lock*) części lub całości plików, celem ochrony ich przed współbieżną modyfikacją.

Metody dostępu

Dostęp do informacji zawartych w pliku może być wykonywany na kilka sposobów, ale najpopularniejsze są dwa. Od metody dostępu do informacji zawartych w pliku zależy efektywność pracy z pamięcią masową. Niektóre systemy operacyjne pozwalają korzystać z tylko jednej metody dostępu do pliku, inne z kilku.

Dostęp sekwencyjny

Dostęp sekwencyjny jest metodą dostępu opartą na taśmowym modelu pliku. Wskaźnik pliku jest zawsze zwiększany, czyli można „poruszać się” jedynie „do przodu” pliku. Większość systemów pozwala zwiększać wskaźnik o jeden, czyli możliwy jest odczyt lub zapis kolejnych bloków alokacji pliku. Istnieją jednak systemy, w których można zwiększać wartość wskaźnika o pewną ustaloną wartość n , która może być większa od jeden.

Dostęp swobodny

W *dostępie swobodnym* zwanym również *dostępem bezpośrednim* wskaźnik pliku może być zwiększany lub zmniejszany o dowolną wielkość. Pozwala to na zapis lub odczyt dowolnego bloku alokacji pliku bez konieczności odczytywania jego poprzedników. Ten sposób jest szybszy od dostępu sekwencyjnego. Dodatkową jego zaletą jest, to że umożliwia również sekwencyjny odczyt/zapis pliku. Wystarczy, że te operacje będą wykonywane na kolejnych blokach alokacji pliku.

Dostęp indeksowy

Ten rodzaj dostępu jest modyfikacją dostępu bezpośredniego. Nadaje się do zastosowania w systemach plików tworzonych na potrzeby baz danych. W *dostępie indeksowym* z każdym dużym plikiem kojarzony jest mały plik, którego zawartość pełni podobną rolę, jak rola indeksu w książkach. Ułatwia ona wyszukiwanie informacji w dużym pliku. W małym pliku zawarte są hasła związane z wyszukiwanymi informacjami oraz numery bloków alokacji dużego pliku, od których rozpoczyna się ciąg bloków w których te informacje są zapisane.

Semantyka spójności

Semantyka spójności nazywana również *semantyką spoistości* lub *semantyką integralności* jest ważnym pojęciem określającym reguły dostępu do plików współdzielonych w systemach wielozadaniowych, dzięki którym zmiany w takich plikach mogą być dokonywane w sposób bezpieczny od sytuacji hazardowych. Semantyka ta określa również zakres wzajemnej widoczności zmian jakich dokonują współbieżnie użytkownicy w pliku. Następne slajdy zawierają opis trzech semantyk spójności, pochodzących z różnych systemów operacyjnych. Wszystkie operacje dokonywane na pliku, wraz z jego otwarciem i zamknięciem będą określane w tych opisach mianem *sesji plikowej*.

Semantyka spójności systemu Unix

Semantyka ta definiuje dwie reguły:

- 1 Jeśli plik jest współdzielony przez kilku użytkowników (np. przez kilka procesów) i jeden z nich dokona modyfikacji zawartości tego pliku, to jej skutki są od razu widoczne dla pozostałych użytkowników.
- 2 Możliwe jest użycie trybu współdzielenia, w którym użytkownicy dzielą wskaźnik pliku. Modyfikacja jego zawartości przez któregoś z użytkowników ma wpływ na operacje wykonywane na pliku przez pozostałych użytkowników.

W semantyce spójności systemu Unix, zasób jakim jest plik ma tylko jeden obraz, widoczny dla każdego z użytkowników. Semantyka ta powoduje opóźnienia w realizacji operacji wykonywanych na pliku, ze względu na konieczność modyfikacji (zapisu) współdzielonego pliku na zasadach wyłączości.

Semantyka spójności systemu Unix

Semantyka ta definiuje dwie reguły:

- ❶ Jeśli plik jest współdzielony przez kilku użytkowników (np. przez kilka procesów) i jeden z nich dokona modyfikacji zawartości tego pliku, to jej skutki są od razu widoczne dla pozostałych użytkowników.
- ❷ Możliwe jest użycie trybu współdzielenia, w którym użytkownicy dzielą wskaźnik pliku. Modyfikacja jego zawartości przez któregoś z użytkowników ma wpływ na operacje wykonywane na pliku przez pozostałych użytkowników.

W semantyce spójności systemu Unix, zasób jakim jest plik ma tylko jeden obraz, widoczny dla każdego z użytkowników. Semantyka ta powoduje opóźnienia w realizacji operacji wykonywanych na pliku, ze względu na konieczność modyfikacji (zapisu) współdzielonego pliku na zasadach wyłączości.

Semantyka spójności systemu Unix

Semantyka ta definiuje dwie reguły:

- ❶ Jeśli plik jest współdzielony przez kilku użytkowników (np. przez kilka procesów) i jeden z nich dokona modyfikacji zawartości tego pliku, to jej skutki są od razu widoczne dla pozostałych użytkowników.
- ❷ Możliwe jest użycie trybu współdzielenia, w którym użytkownicy dzielą wskaźnik pliku. Modyfikacja jego zawartości przez któregoś z użytkowników ma wpływ na operacje wykonywane na pliku przez pozostałych użytkowników.

W semantyce spójności systemu Unix, zasób jakim jest plik ma tylko jeden obraz, widoczny dla każdego z użytkowników. Semantyka ta powoduje opóźnienia w realizacji operacji wykonywanych na pliku, ze względu na konieczność modyfikacji (zapisu) współdzielonego pliku na zasadach wyłączości.

Semantyka sesji plikowej

Ta semantyka została zastosowana po raz pierwszy w systemie Andrew. Definiowała ona następujące reguły:

- 1 Jeśli kilku użytkowników ma otwarty plik współdzielony i jeden z nich dokona modyfikacji jego zawartości, to jej skutki nie są widoczne natychmiast dla pozostałych.
- 2 Zmiany w pliku są widoczne tylko wtedy, gdy użytkownik, który ich dokonał zamknie plik, a potem pozostali dokonają zamknięcia i otwarcia tego pliku.

Te reguły wymagają, aby każdy z użytkowników pliku otrzymywał jego kopię, unikatowy obraz tego pliku dostępny dla każdego z nich osobno.

Semantyka sesji plikowej

Ta semantyka została zastosowana po raz pierwszy w systemie Andrew. Definiowała ona następujące reguły:

- ❶ Jeśli kilku użytkowników ma otwarty plik współdzielony i jeden z nich dokona modyfikacji jego zawartości, to jej skutki nie są widoczne natychmiast dla pozostałych.
- ❷ Zmiany w pliku są widoczne tylko wtedy, gdy użytkownik, który ich dokonał zamknie plik, a potem pozostali dokonają zamknięcia i otwarcia tego pliku.

Te reguły wymagają, aby każdy z użytkowników pliku otrzymywał jego kopię, unikatowy obraz tego pliku dostępny dla każdego z nich osobno.

Semantyka sesji plikowej

Ta semantyka została zastosowana po raz pierwszy w systemie Andrew. Definiowała ona następujące reguły:

- 1 Jeśli kilku użytkowników ma otwarty plik współdzielony i jeden z nich dokona modyfikacji jego zawartości, to jej skutki nie są widoczne natychmiast dla pozostałych.
- 2 Zmiany w pliku są widoczne tylko wtedy, gdy użytkownik, który ich dokonał zamknie plik, a potem pozostali dokonają zamknięcia i otwarcia tego pliku.

Te reguły wymagają, aby każdy z użytkowników pliku otrzymywał jego kopię, unikatowy obraz tego pliku dostępny dla każdego z nich osobno.

Semantyka niezmiennych plików współdzielonych

Tę semantykę stosuje się głównie w systemach rozproszonych. Jeśli plik jest współdzielony, to możliwy jest tylko jego odczyt, nie można zaś zmodyfikować jego zawartości. Tak, jak zazwyczaj, plik jest udostępniony użytkownikom za pomocą jego nazwy, ale w tym wypadku nazwa odnosi się nie do pliku w tradycyjnym znaczeniu, ale do jego określonej zawartości, która nie może podlegać zmianom. Jedyną możliwością dokonania modyfikacji zawartości pliku dla użytkownika jest wykonanie jej kopii, wprowadzenie zmian i opublikowanie tej treści pod inną nazwą.

Katalogi fizyczne i logiczne

Na poprzednim wykładzie wprowadzone zostało pojęcie katalogu fizycznego, czyli katalogu głównego urządzenia. Ponieważ jego pojemność (liczba wpisów) jest ograniczona, to w systemach operacyjnych są definiowane *katalogi logiczne*. Katalog logiczny jest strukturą, której zawartość stanowią wykazy informacji o innych plikach². Informacje o plikach to inaczej ich *metadane*. Tradycyjnie katalogi logiczne najczęściej są implementowane jako pliki, choć część systemów operacyjnych stosuje inne rozwiązania. Do metadanych plików umieszczonych w katalogach mogą zaliczać się: unikatowa *nazwa pliku*, informacja o *typie pliku* (np.: trzyliterowe rozszerzenie nazwy), *wskaźnik pliku*, *rozmiar pliku* (w bajtach lub jednostkach alokacji), *lokalizacja* pliku, czyli np. numer pierwszego bloku alokacji przedzielonego plikowi, *informacje o ochronie*, czyli o trybie dostępu, *licznik użycia*, określający ile istnieje tzw. dowiązań do pliku, *identyfikator właściciela i/lub twórcy pliku* oraz *czas utworzenia, ostatniej modyfikacji i/lub ostatniego dostępu* do pliku. Katalogi tworzą *strukturę katalogów*. Taka struktura może obejmować nie tylko jedno urządzenie fizyczne, ale również kilka takich urządzeń (np. struktura katalogów systemu Unix). Dzięki temu system operacyjny ukrywa przed użytkownikiem faktyczną lokalizację plików.

²Wpisy tego samego typu co w katalogu fizycznym.

Wykaz pozycji

Katalog jest zbiorem pozycji, które zawierają metadane plików. Od tego w jaki sposób zostanie ten wykaz pozycji zaimplementowany zależy efektywność odnajdywania informacji o plikach, a także usuwania i tworzenia plików. Oto niektóre z możliwych realizacji:

- **Tablica**-wymaga liniowego czasu przeszukiwania. Pozycja pliku, który został usunięty może być oznaczana w specjalny sposób jako pusta i później użyta do utworzenia nowego pliku. Jeśli nie ma pustych pozycji, to informacje o nowym pliku są umieszczane w pierwszej wolnej pozycji wykazu. Aby przyspieszyć wyszukiwanie można zastosować algorytm wyszukiwania binarnego, ale to wymaga utrzymywania wykazu plików w stanie posortowanym.
- **Lista powiązana**-również wymaga liniowego czasu przeszukiwania, ale operacje wstawiania i usuwania pozycji we wpisie są łatwiejsze do przeprowadzenia, w szczególności łatwiej jest utrzymywać wykaz w stanie posortowania.
- **Drzewo binarne powiązane**-wymaga logarytmicznego czasu przeszukiwania
- **Tablica z haszowaniem**(ang. *hash table*)-pozwala na szybsze wyszukiwanie, jak również tworzenie i usuwanie plików, niż zwykła tablica. Problemem jest jednak zmiana funkcji haszującej, której postać jest uzależniona od wielkości tej struktury.

Wykaz pozycji

Katalog jest zbiorem pozycji, które zawierają metadane plików. Od tego w jaki sposób zostanie ten wykaz pozycji zaimplementowany zależy efektywność odnajdywania informacji o plikach, a także usuwania i tworzenia plików. Oto niektóre z możliwych realizacji:

- **Tablica**-wymaga liniowego czasu przeszukiwania. Pozycja pliku, który został usunięty może być oznaczana w specjalny sposób jako pusta i później użyta do utworzenia nowego pliku. Jeśli nie ma pustych pozycji, to informacje o nowym pliku są umieszczane w pierwszej wolnej pozycji wykazu. Aby przyspieszyć wyszukiwanie można zastosować algorytm wyszukiwania binarnego, ale to wymaga utrzymywania wykazu plików w stanie posortowanym.
- **Lista powiązana**-również wymaga liniowego czasu przeszukiwania, ale operacje wstawiania i usuwania pozycji we wpisie są łatwiejsze do przeprowadzenia, w szczególności łatwiej jest utrzymywać wykaz w stanie posortowania.
- **Drzewo binarne powiązane**-wymaga logarytmicznego czasu przeszukiwania
- **Tablica z haszowaniem**(ang. *hash table*)-pozwala na szybsze wyszukiwanie, jak również tworzenie i usuwanie plików, niż zwykła tablica. Problemem jest jednak zmiana funkcji haszującej, której postać jest uzależniona od wielkości tej struktury.

Wykaz pozycji

Katalog jest zbiorem pozycji, które zawierają metadane plików. Od tego w jaki sposób zostanie ten wykaz pozycji zaimplementowany zależy efektywność odnajdywania informacji o plikach, a także usuwania i tworzenia plików. Oto niektóre z możliwych realizacji:

- **Tablica**-wymaga liniowego czasu przeszukiwania. Pozycja pliku, który został usunięty może być oznaczana w specjalny sposób jako pusta i później użyta do utworzenia nowego pliku. Jeśli nie ma pustych pozycji, to informacje o nowym pliku są umieszczane w pierwszej wolnej pozycji wykazu. Aby przyspieszyć wyszukiwanie można zastosować algorytm wyszukiwania binarnego, ale to wymaga utrzymywania wykazu plików w stanie posortowanym.
- **Lista powiązana**-również wymaga liniowego czasu przeszukiwania, ale operacje wstawiania i usuwania pozycji we wpisie są łatwiejsze do przeprowadzenia, w szczególności łatwiej jest utrzymywać wykaz w stanie posortowania.
- **Drzewo binarne powiązane**-wymaga logarytmicznego czasu przeszukiwania
- **Tablica z haszowaniem**(ang. *hash table*)-pozwala na szybsze wyszukiwanie, jak również tworzenie i usuwanie plików, niż zwykła tablica. Problemem jest jednak zmiana funkcji haszującej, której postać jest uzależniona od wielkości tej struktury.

Wykaz pozycji

Katalog jest zbiorem pozycji, które zawierają metadane plików. Od tego w jaki sposób zostanie ten wykaz pozycji zaimplementowany zależy efektywność odnajdywania informacji o plikach, a także usuwania i tworzenia plików. Oto niektóre z możliwych realizacji:

- **Tablica**-wymaga liniowego czasu przeszukiwania. Pozycja pliku, który został usunięty może być oznaczana w specjalny sposób jako pusta i później użyta do utworzenia nowego pliku. Jeśli nie ma pustych pozycji, to informacje o nowym pliku są umieszczane w pierwszej wolnej pozycji wykazu. Aby przyspieszyć wyszukiwanie można zastosować algorytm wyszukiwania binarnego, ale to wymaga utrzymywania wykazu plików w stanie posortowanym.
- **Lista powiązana**-również wymaga liniowego czasu przeszukiwania, ale operacje wstawiania i usuwania pozycji we wpisie są łatwiejsze do przeprowadzenia, w szczególności łatwiej jest utrzymywać wykaz w stanie posortowania.
- **Drzewo binarne powiązane**-wymaga logarytmicznego czasu przeszukiwania
- **Tablica z haszowaniem**(ang. *hash table*)-pozwala na szybsze wyszukiwanie, jak również tworzenie i usuwanie plików, niż zwykła tablica. Problemem jest jednak zmiana funkcji haszującej, której postać jest uzależniona od wielkości tej struktury.

Wykaz pozycji

Katalog jest zbiorem pozycji, które zawierają metadane plików. Od tego w jaki sposób zostanie ten wykaz pozycji zaimplementowany zależy efektywność odnajdywania informacji o plikach, a także usuwania i tworzenia plików. Oto niektóre z możliwych realizacji:

- **Tablica**-wymaga liniowego czasu przeszukiwania. Pozycja pliku, który został usunięty może być oznaczana w specjalny sposób jako pusta i później użyta do utworzenia nowego pliku. Jeśli nie ma pustych pozycji, to informacje o nowym pliku są umieszczane w pierwszej wolnej pozycji wykazu. Aby przyspieszyć wyszukiwanie można zastosować algorytm wyszukiwania binarnego, ale to wymaga utrzymywania wykazu plików w stanie posortowanym.
- **Lista powiązana**-również wymaga liniowego czasu przeszukiwania, ale operacje wstawiania i usuwania pozycji we wpisie są łatwiejsze do przeprowadzenia, w szczególności łatwiej jest utrzymywać wykaz w stanie posortowania.
- **Drzewo binarne powiązane**-wymaga logarytmicznego czasu przeszukiwania
- **Tablica z haszowaniem**(ang. *hash table*)-pozwala na szybsze wyszukiwanie, jak również tworzenie i usuwanie plików, niż zwykła tablica. Problemem jest jednak zmiana funkcji haszującej, której postać jest uzależniona od wielkości tej struktury.

Katalog jednopoziomowy

Katalog jednopoziomowy jest najprostszym przykładem struktury katalogowej. Do jej realizacji wystarczy jedynie katalog urządzenia. Ponieważ wszystkie pliki w katalogu muszą mieć unikatową nazwę, a dodatkowo część systemów operacyjnych ogranicza długość nazwy pliku, to katalogi jednopoziomowe mają ograniczone zastosowanie. W szczególności nie mogą one być stosowane w systemach wielodostępnych. Zastosowanie takiej struktury w obecnych systemach jest bardzo rzadkie.

Katalog dwupoziomowy

Struktura katalogu dwupoziomowego pozwala na tworzenie katalogów logicznych, które pełnią rolę *katalogów domowych* użytkowników (ang. *home directory*). Tak więc każdy użytkownik posiada własny katalog, w którym może przechowywać własne pliki. Wymóg unikalności nazwy pliku obowiązuje tylko w obrębie katalogu, a więc jeśli jeden z użytkowników nazwie swój plik, tak jak inny użytkownik, to nie dochodzi do konfliktu nazw. Katalog dwupoziomowy pozwala również korzystać użytkownikom z plików innych użytkowników. Wystarczy jeśli poprzedzą oni nazwę pliku nazwą katalogu użytkownika, w którym ten plik się znajduje. Taka konstrukcja (nazwa katalogu razem z nazwą pliku) nazywa się *ścieżką* (ang. *path*). Oprócz katalogów użytkowników w katalogu głównym tworzony jest katalog zawierający oprogramowanie systemowe. Aby uniknąć konieczności podawania przez użytkowników ścieżki do plików znajdujących się w katalogu systemowym w systemach operacyjnych stosuje się *ścieżkę wyszukiwania* (ang. *search path*), która określa kolejność w jakiej system przeszukuje katalogi w poszukiwaniu żadanego przez użytkownika pliku. Jeśli w tej ścieżce na pierwszej pozycji wymieniony jest katalog systemowy, to jest on najpierw przeszukiwany, a następnie jest przeszukiwany katalog użytkownika.

Drzewo katalogów

Aby użytkownik mógł posiadać pewną liczbę plików o takich samych nazwach oraz aby mógł lepiej zarządzać swoimi plikami należy pozwolić mu na zakładanie własnych katalogów, które mogą tworzyć wielopoziomową strukturę. Ta struktura ma postać drzewa. W *drzewie katalogów* występuje pojęcie *katalogu bieżącego* (ang. *working directory* lub *current directory*). Jest to katalog, który jest w danej chwili używany przez użytkownika (np. proces). Użytkownik ma możliwość poruszania się po strukturze katalogów, gdyż każdy katalog ma dwie specjalne pozycje, które stanowią dowiązanie do samego siebie (nazywane najczęściej za pomocą jednej kropki) i dowiązanie do katalogu znajdującego się bezpośrednio wyżej w hierarchii (nazywane najczęściej za pomocą dwóch kropek). W drzewie katalogów występuje również pojęcie *ścieżki względnej*, która określa położenie pliku względem katalogu bieżącego i *ścieżki bezwzględnej* określającej położenie pliku względem katalogu głównego. Stosowane są dwa podejścia odnośnie usuwania katalogów. W pierwszym zakłada się, że jeśli użytkownik chce skasować katalog, to jego zawartość, wraz z podkatalogami również należy skasować i wykonuje się ich rekurencyjne usuwanie. W drugim podejściu system odmawia skasowania katalogu jeśli nie jest on pusty.

Graf katalogów

Aby ułatwić użytkownikom współdzielenie pliku, czyli współbieżny dostęp do jego zawartości, wprowadzane są *dowiązania* (ang. *link*). Wprowadzenie dowiązań przekształca drzewo katalogów w *graf katalogów*. Istnieją dwa rodzaje dowiązań: *dowiązania sztywne lub twarde* (ang. *hard links*) i *dowiązania symboliczne* (ang. *symbolic links*). Tworzenie dowiązania twardego sprowadza się do skopiowania pozycji z katalogu źródłowego do docelowego. Zarówno oryginał jak i kopia są nie do odróżnienia, co może sprawiać problemy przy przeszukiwaniu katalogu (niekończące się pętle) i przy usuwaniu pliku. Jeśli usuniemy plik za pomocą dowiązania, to zostaje oryginalny wpis, odwrotna sytuacja również jest możliwa. Częściowym rozwiązaniem tego problemu może być związanie z zawartością pliku *wykazu dowiązań* lub *licznika dowiązań* i usuwanie oryginalnego pliku tylko wtedy gdy taki wykaz jest pusty lub gdy licznik dowiązań jest równy zero. Wymienionych wad są w większości pozbawione dowiązania symboliczne, które są specjalnymi pozycjami w katalogu będącymi wskaźnikami na nazwę oryginalnego pliku. Dzięki temu można wypisać nazwę dowiązania przy przeszukiwaniu katalogu, ale go nie *rozwiązywać* (ang. *resolve*), co zapobiega powstawaniu nieskończonych pętli. Operacja usunięcia dowiązania niszczy tylko dowiązanie, oryginalny plik pozostanie w systemie plików. Problem usunięcia pliku jest również częściowo rozwiązany - jeśli zniknie pozycja pliku, to również jego nazwa, na którą wskazuje dowiązanie. Wielu problemów unika się jeśli graf katalogów jest acykliczny. Można to zapewnić wykonując operacje wykrywania odwołań cyklicznych połączoną z odśmiecaniem (ang. *garbage collection*). Jest to jednak czasochłonna operacja.

Ochrona plików

Są dwa główne cele ochrony plików - ochrona przed zniszczeniem, którą zapewnia się wprowadzając mechanizmy automatycznego tworzenia kopii zapasowych oraz zapewnienie poufności ich treści. To ostatnie zadanie wymaga zapewnienia kompromisu pomiędzy możliwością współdzielenia plików przez użytkowników, a koniecznością ograniczenia dostępu do nich. Ograniczenia te dotyczą operacji wykonywanych na plikach (odczyt, zapis, wykonanie, modyfikacja, dopisywanie) oraz operacji wykonywanych na katalogach (przeglądanie, poruszanie się po strukturze katalogów, usuwanie i tworzenie plików). Ochrona może dotyczyć zawartości pliku lub ścieżki dostępu do pliku. Częściej stosuje się to drugie rozwiązanie. Istnieje również wiele sposobów implementacji tej ochrony.

Nazywanie

Najprostszy sposób ochrony plików polega na zakazaniu wyświetlania nazw plików znajdujących się w katalogach. Wówczas ma do nich dostęp jedynie użytkownik, który te nazwy zna. Ten rodzaj ochrony jest niewygodny i zawodny. Ponieważ długość nazw plików jest ograniczana przez systemy, a użytkownicy tworzą nazwy mnemotechniczne, to łatwo jest intruzowi taką nazwę ustalić.

Hasła

Pliki mogą być chronione przez system *hasła*. Jest on bardziej bezpieczny, niż ukrywanie nazw, ale posiada swoje wady. Jeśli z każdym plikiem zwiążemy hasło, to liczba haseł, którą musi zapamiętać użytkownik staje się bardzo duża. Jeśli z drugiej strony zastosujemy jedno hasło do wszystkich plików, lub do wydzielonej grupy plików, to złamanie tego hasła umożliwi intruzowi dostęp do dużej ilości informacji zawartej w systemie. Ochrona za pomocą haseł nie pozwala również na ograniczenie dostępu do pliku. Albo dajemy użytkownikowi całkowity dostęp do zawartości pliku, albo w ogóle go zabraniamy.

Wykaz dostępów

Z każdym plikiem lub katalogiem można związać *wykaz dostępów* (ang. *access control list*), czyli opis praw jakie przysługują poszczególnym użytkownikom do tego pliku (czy użytkownik może dokonywać zapisu, odczytu itd.) po ich uwierzytelnieniu przez system operacyjny. Jeśli użytkownik próbuje wykonać jakąś operację, to system operacyjny odnajduje jego identyfikator w wykazie dostępów i sprawdza jakie prawa mu przysługują. Jeśli w wykazie znajduje się operacja, którą użytkownik chce wykonać, to jest ona wykonywana, w przeciwnym razie system odmawia dostępu użytkownikowi. Wadą tego rozwiązania może okazać się wielkość takiego wykazu.

Dostęp grupowy

W *dostępie grupowym* z każdym plikiem lub katalogiem związanych jest dziewięć bitów. Każda trójka bitów opisuje prawa dostępu do pliku pozwalające na odczyt, zapis i wykonanie (rwx w notacji uniksowej). Również znaczenie każdej z tych trójek jest inne. Pierwsza opisuje prawa właściciela pliku (user w notacji uniksowej), druga prawa grupy użytkowników do której należy właściciel pliku (group w notacji uniksowej), a trzecia określa prawa dostępu dla pozostałych użytkowników w systemie (other w notacji uniksowej). Grupy użytkowników są zdefiniowane w specjalnie chronionym pliku systemowym. Dzięki *dostępowi grupowemu* możliwe jest współdzielenie plików. Również ilość informacji związanych z ochroną plików, które trzeba zapamiętać jest mała w stosunku do wykazuostępów.

Pytania

?

Koniec

Dziękuję Państwu za uwagę!